Part 2. Machine Learning

Tia C

(I am severely dyscalculic, so the maths and formulas in this report may be wrong.)

1. Introduction

This investigation aims to identify the most suitable machine learning algorithm to improve the company's Intrusion Detection System (IDS). The IDS that is currently in place is working and gathering data for analysis, however the team cannot digest the raw data. The machine learning algorithm will be used to process the data and feed the information back to the team in a more accessible format. By using machine learning as a tool with the IDS system, there will be lower false alarms and a higher detection rate.

The data that the machine learning algorithm will be trained on is currently packets that have been captured from the network. These packets have information such as the service and protocol being used, as well as attack categories. The attack category will be used to categorise the packet data through the algorithm. There are ten attack categories, they have been split into normal and abnormal traffic in Table 1.

Normal	Abnormal
Normal	Fuzzers
Analysis	Backdoors
Generic	DoS
	Exploits
	Reconnaissance
	Shellcode
	Worms

Table 1 - List of attack categories, split into Normal and Abnormal traffic

2. Machine Learning and Intrusion Detection Systems

Machine Learning is used in many industries to help advance their technology and software. Machine learning is the process of using algorithms to find patterns in data, whilst learning from the data being passed into it. It eventually can make predictions from the data, after it has learned from a training set. There are three types of learning within machine learning, Supervised, Unsupervised and Reinforcement. For brevity, the output has been labelled as Normal or Abnormal Traffic.

Supervised learning is where the data is labelled, so the algorithm can identify the patterns and labels. The training set that the company is using has the attack categories labelled, the algorithm can use this training set to identify patterns that may not be obvious to a human. It can then use this when it is making predictions about the data, Figure 1 shows diagram of the process. For example, if the algorithm was being trained on identifying if a review was positive or negative, the training set would have positive reviews and negative reviews – the algorithm will learn to predict the review's sentiment by the words used and will then decide if it is a negative or positive sentiment.

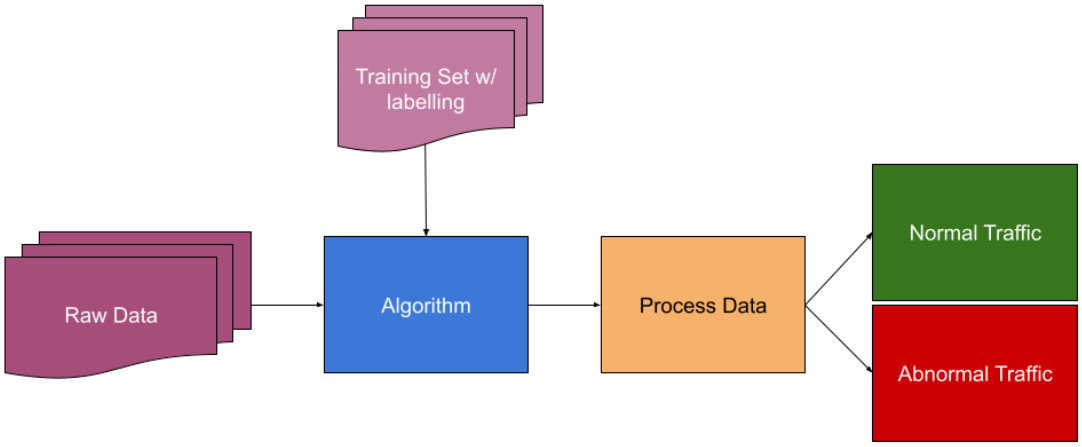

Figure 1 - Diagram demonstrating Supervised Learning Process for the IDS

Unsupervised learning is where there are no labels, the algorithm must identify patterns within the data by itself. The algorithm will then categorise similar pieces of data together. Unsupervised learning is applied in many cybersecurity applications as it can adapt to malicious situations more effectively. Figure 2 below, shows a diagram of this process. As before in the process data stage, it will be categorised into attacks where the code that is separate from the model will flag the packet as normal or abnormal dependent on the model's prediction. For example, if an attacker is trying different variations of an attack or exploit, the algorithm will pick up on this abnormal data and flag it as suspicious.

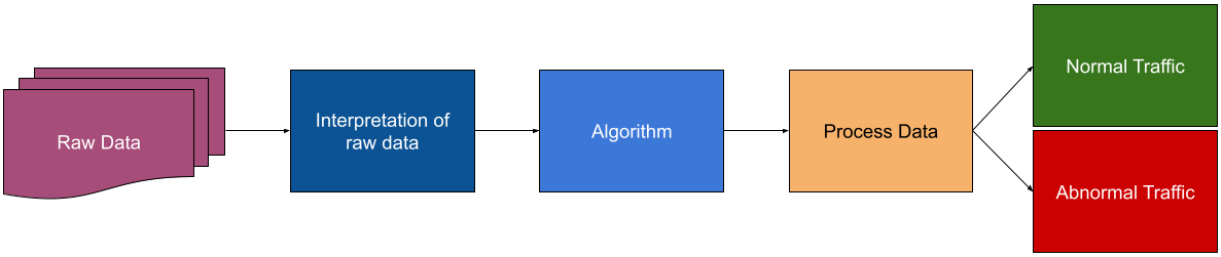

Figure 2 - Diagram demonstrating Unsupervised Learning Process for the IDS

Reinforcement learning is where the model is either positively or negatively reinforced in response to a behaviour or action. This method of learning is different from the previous types of learning, as it will change the output of its current input depending on the output of the previous output. The diagram demonstrating this process can be seen in figure 3. It is like a dog being taught a trick. When the dog is told to sit, and successfully sits – it is given praise and a treat. If the dog is told to sit and does not – the dog will not be given praise or a treat. The dog will soon associate the word with the behaviour as it is rewarded for the behaviour when it is told to carry it out.

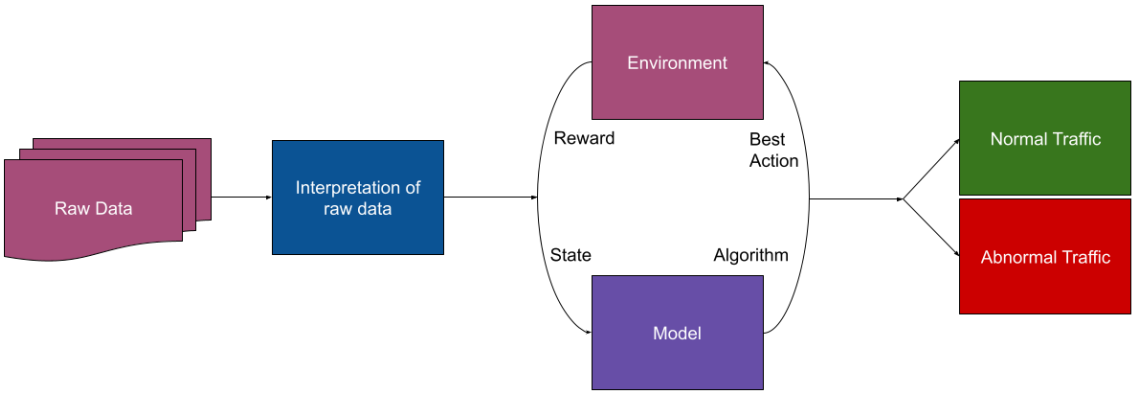

Figure 3 - Diagram demonstrating Reinforced Learning Process for the IDS

This investigation will look at a Supervised and an Unsupervised algorithm. The algorithms that will be discussed are the Random Forest algorithm (Supervised) and K-Means Clustering (Unsupervised).

3. Random Forest – Supervised Method

The random forest algorithm is a supervised method, so must make use of the training dataset provided by the IDS team. The algorithm makes use of multiple decision trees, to create the forest and using a 'bagging' method then compares the output of the decision trees to determine the most accurate overall output. If the data is abnormal, this will be flagged to an analyst to decide on the best possible course of action. For example, if there are five decision trees and three of the five determine that the packet is classified as a generic packet, with the other two decision trees classifying it a DoS – it will classify the packet as generic.

Bagging or Bootstrap Aggregating as it formally known is used to ensure that the predictions formed are accurate. This means it is less likely that false positives and negatives to occur. Using the training set as an example, bagging is carried out by randomly selecting packets from the dataset. The number of packets selected must be the same size as the dataset, so the original and bootstrap dataset will have 81 packets. From the bootstrap datasets, features are randomly selected to create a new training dataset for each of the trees in the forest. An example can be found below in Figure 4.

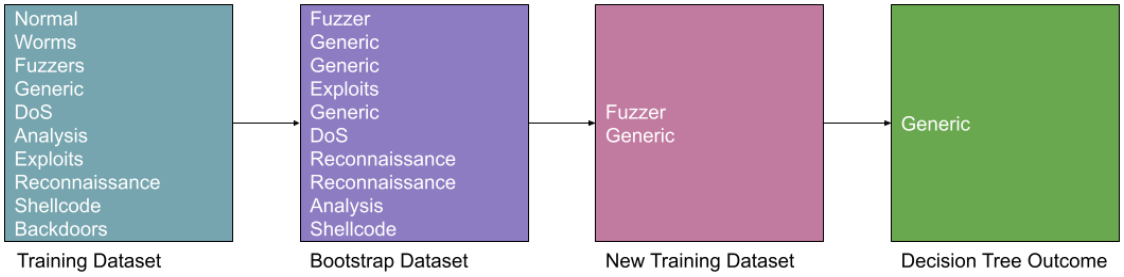

Figure 4 - Example of random forest method with IDS training dataset

Random forest would be a suitable option, as it is a very accurate algorithm and can handle many input variables meaning that it is suitable to work with large datasets, such as the dataset that the IDS team will be using. Random Forest can also be used for prediction of missing data, which in turn means that it will be accurate even if there is data missing from the dataset. The algorithm is easy and quick to apply and due to its nature of learning, it is immune to over-training.

However, there are limitations to the algorithm. The algorithm may be immune to over-training but is not immune to over-fitting. Over-fitting is when the algorithm begins to learn the pattern and

details of the training dataset, to the extent that it cannot learn the pattern and details of the data being ingested in production. This results in a negative performance by the algorithm. As well as this, random forests tend to require more processing as each decision tree needs to be processed. This means that the more trees in the forest, the more processing required.

4. Multi-Layer Perceptron - Neural Network

The Multi-Layer Perceptron is also a supervised method but is also an artificial neural network. The neural network works in the same way that brains work. There are multiple types of neural network, including a single layer neural network. The multi-layer perceptron has input, hidden and output neurons. These neurons have specific purposes, the input neurons receive the data that is to be ingested, the hidden neurons are where the machine processes the data and does the learning that is required. Lastly there is the output neurons where the information is outputted. This is shown in figure 2 below.

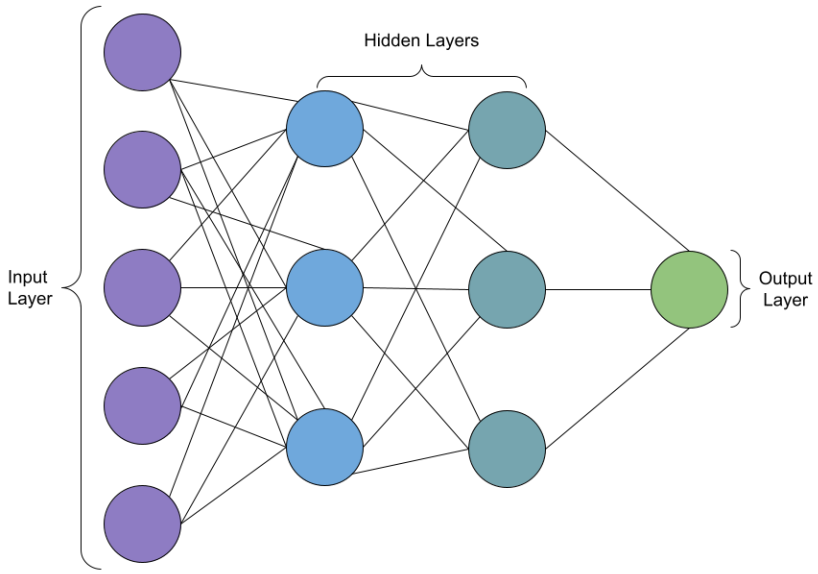

Figure 5 - Example of a Multi-Layer Perceptron Neural Network

The multi-layer perceptron has multiple layers of hidden neurons. These layers are 'stacked' on top of each other, forming a feature hierarchy. The first layer of the hidden neurons will process the input layer, the second layer of the hidden neurons will then process the input of the first hidden neuron layer and so on. The machine will continue processing the data and learning until it reaches the output layer. The links or 'synapses' between layers are called weights and are random values between 0 and 1. These weights are used within the calculations and decisions; the weights are adjusted when the output of the environment is observed, to become more accurate. This is how the machine learns with multi-layer perceptron neural network.

The multi-layer perception would also be a possible option, as it is able to take complex inputs and output this in a user-friendly way. The algorithm could be taught to identify which attacks are more severe and should be flagged immediately, whilst identifying which attacks may just be reconnaissance and can either be blocked or flagged for review by a human if required. The neural network is robust to noisy datasets and has a higher fault tolerance too.

However, there are limitations with using a neural network – particularly that it takes time to train the algorithm. Neural networks are also not verbose in explaining how they reached an outcome;

this may prove difficult with the IDS if it is flagging suspicious packets but cannot explain the reasoning behind the flag. The time it takes to teach the neural network is also very long and may prove to be difficult to implement into the system.

5. Design of classifier

The author recommends that the algorithm that should be implemented is the Random Forest. The reason that this algorithm should be used is that the advantages outweigh the disadvantages. The possible limitation of overfitting can also be overcome by 'pruning' some of the trees within the forest. The algorithm can be applied almost immediately, which is more efficient than the neural network. The dataset being used to train is two-dimensional and is labelled, which makes the dataset suitable for the Random Forest algorithm.

To implement the model, the value to be predicted needs to be labelled. This is done by setting the label in the script, for this dataset the label would be the attack category. A baseline error should also be calculated to ensure that the model is making accurate predictions, if the average baseline is not achieved then the algorithm may not be effective. To further ensure that the model is making accurate predictions, a testing dataset without the attack categories in should be created. Once the algorithm has been trained, the testing dataset should be used and compared to the training dataset.

Before the model is fitted, the data must be normalised to ensure that the algorithm is giving accurate predictions. Feature scaling is used to ensure that features in the dataset contribute evenly to the overall prediction. To begin the learning stage, the random forest classification model from scikit should be used. The default number of trees is 100, so this may be used. It may require trial and error to begin with to find the suitable number of trees. To communicate the results a bar chart showing the attack category of the packets may be suitable for analysts to see the volume of malicious attempts in the network each day, whilst a line chart may also be suitable to see when and where these abnormal packets are being sent.

6. Evaluation Metrics

When the model has been trained and tested, it is important to ensure that the model is producing accurate predictions. The metrics that can be used to evaluate the accuracy of the model's predictions are the Confusion Matrix and the Classification accuracy. The confusion matrix can be used to see where the model made accurate predictions and where the model was unable to make correct predictions. This can be used to understand the working of the model and where it may need more support in learning. Scikit has a confusion_matrix model that should be used for this purpose.

The classification report is generated by scikit learn and considers the precision, recall, F1 score and accuracy. The precision is the number of predictions that were true positives whilst the recall is the number of true positives that were identified. The calculations used to obtain the precision and recall can be seen below.

$$\textbf{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad \textbf{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

The F1 score is a mean of both the precision and recall. These are used to inspect the accuracy of each attack category predicted. The Confusion Matrix should be used in conjunction with this to evaluate the accuracy of the model as a whole and by each attack category.

References

Akkaya, Berke & Çolakoğlu, Nurdan. (2019). Comparison of Multi-class Classification Algorithms on Early Diagnosis of Heart Diseases.

Baeldung CS. 2020. *Advantages and Disadvantages of Neural Networks*. [online] Available at: <https://www.baeldung.com/cs/neural-net-advantages-disadvantages>

GeeksforGeeks. 2021. *ML | Feature Scaling – Part 2 - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/ml-feature-scaling-part-2/> [Accessed 23 April 2022].

GeeksforGeeks. 2022. *Reinforcement learning - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/what-is-reinforcement-learning/> [Accessed 11 April 2022].

Medium. 2020. *Evaluating a Random Forest model*. [online] Available at: <https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56> [Accessed 23 April 2022].

Medium. 2022. *Understanding Random Forest*. [online] Available at: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> [Accessed 11 April 2022].

MIT Technology Review. 2018. *What is machine learning?*. [online] Available at: <https://www.technologyreview.com/2018/11/17/103781/what-is-machine-learning-we-drew-you-another-flowchart/> [Accessed 11 April 2022].

scikit-learn. 2022. *sklearn.metrics.confusion_matrix*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html> [Accessed 23 April 2022].

Sharkawy, Abdel-Nasser. (2020). Principle of Neural Network and Its Main Types: Review. Journal of Advances in Applied & Computational Mathematics. 7. 8-19. 10.15377/2409-5761.2020.07.2.

Simplilearn. 2022. *The Complete Guide on Overfitting and Underfitting in Machine Learning*. [online] Available at: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/overfitting-and-underfitting> [Accessed 23 April 2022].

TechNative. 2021. *Why Unsupervised Machine Learning is the Future of Cybersecurity*. [online] Available at: <https://technative.io/why-unsupervised-machine-learning-is-the-future-of-cybersecurity/> [Accessed 11 April 2022].

Tesfahun and Bhaskari, "Intrusion Detection Using Random Forests Classifier with SMOTE and Feature Reduction," *2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*, 2013, pp. 127-132, doi: 10.1109/CUBE.2013.31.